

Adatbázis alapú rendszerek

2023-2024/2
IB152L-10

Quiz

Készítette:
Kávai Illés
Kaszás Zsolt József
Velicán-Benkő Attila

Munka felosztása

Közösen dolgozunk a feladatok megoldásán.

Értékelési mód:

A csoport tagjai közösen dolgoznak, minden tag ugyanannyi pontot kap.

Feladat szöveges leírása

A weboldal egy kvíz játékot kínál, amely lehetőséget biztosít a felhasználóknak saját szobák létrehozására. Ezekben a szobákban a felhasználók saját kérdéseket állíthatnak be, valamint válaszokat adhatnak hozzájuk. Más játékosok is csatlakozhatnak ezekhez a szobákhoz, és megmérkőzhetnek egymással, hogy mennyi kérdésre tudnak válaszolni az adott szobában.

A weboldal PHP-ban készül, az Oracle 12.0.0.0.1 adatbázist és a SQLDevelopert használjuk az adatbázis létrehozásához és kezeléséhez. A dokumentáció elkészítéséhez a Word programot választottuk. Az alkalmazás fejlesztése a Visual Studio Code/PHPStorm fejlesztői környezetben történik. Diagramok készítésére a ClickCharts-t használjuk.

Az adatbázis tartalmazza a következő hét fő adattáblát: felhasználó, eredmény, kérdés, tema, válasz, szoba, szoba_kerdesei. Ezeket az adattáblákat az alkalmazás kezeli és frissíti.

Az alkalmazás belépő képernyővel indul ahol látható az adatbázishoz kapcsolat sikeressége és egy menüsor amivel tud navigálni más oldalakra. A felhasználó bejelentkezhet a felhasználói fiókjába ehhez a bejelentkezési űrlapot kell kitöltenie. Ha a felhasználó még nem regisztrált, akkor a regisztrációs űrlapot kell kitöltenie. Sikeres bejelentkezés után a felhasználó a főmenübe kerül, a menüsorban megjelenik a Szobák, Főoldal, Kijelentkezés lehetőség.

A szobába való belépést követően lehetőség van szobához való csatlakozásra, a felhasználó által létrehozott szobák törlésére és jelszó módosítására és új szoba létrehozására.

A szobába lépést követően a létrehozója és az admin képes kérdéseket adni és törölni. Egyéb esetben a quiz kitöltéséhez kerül a felhasználó.

A quiz kitöltésekor a szobához tartozó kérdések egyszerre megjelennek, minden válaszhoz megjelenik a kérdés és el kell dönteni, hogy igaz vagy hamis az állítás, ha üresen hagyjuk az a hamisat, ha bejelöljük az az igazat jelenti. Lehetőség van a kérdésre adott válaszokat menteni. A mentést követően átirányít az oldal a Szoba eredményei oldalra.

A Szoba eredményei oldalon látjuk a Toplistát, Adatokat, A Felhasználó legutóbbi eredményét és a Felhasználó átlagpontszámát.

A játékmenet során a játékosok a kvíz kitöltésével versenyeznek, eredményeiket az adatbázisban tárolják. A játék során a játékosok különböző kérdésekre válaszolnak, amelyekhez a program az adattáblákból olvassa be a válaszokat, és a válaszok helyességét is az adatbázisban tárolja.

Követelménykatalógus

Cél: Az alkalmazás célja, hogy interaktív és szórakoztató módon lehetőséget biztosítson a felhasználóknak szobákban való versengésre, ahol különböző témákhoz kapcsolódó kérdésekre válaszolhatnak.

Felhasználó belépése felhasználónév és jelszó segítségével:

A felhasználók beléphetnek a felhasználói fiókjukba felhasználónév és jelszó megadásával.

- **Regisztráció:**

A felhasználóknak lehetőségük van regisztrálni az alkalmazásba, ahol meg kell adniuk néhány alapvető információt, mint például felhasználónév, jelszó, email stb.

- **Játékszobákhoz való csatlakozás:**

A játékosoknak lehetőségük van csatlakozni más játékosok által létrehozott játékszobákhoz, ha ismerik a szoba nevét és jelszavát.

- **Új játékszobák létrehozása:**

A játékosoknak lehetőségük van új játékszobákat létrehozni, amelyeknek meg kell adniuk a szoba nevét és jelszavát.

- **Kérdésekre való válaszadás:**

A játékosoknak lehetőségük van válaszolni a szobához tartozó kérdésekre a játékszobákban.

- **Pontszám szerzése helyes válaszokért:**

A játékosok pontokat szereznek a helyes válaszokért, amelyeket az alkalmazás rögzít és megjelenít.

- Ha egy felhasználó törlésre kerül a rendszerből, és ehhez tartozik legalább egy játékszoba, akkor az ehhez a felhasználóhoz kapcsolódó szobákhoz tartozó felhasználói azonosítók automatikusan NULL értékre állnak.
- Az adminisztrátornak lehetősége van egy felhasználói fiókhoz üres felhasználónevet rendelni, amely lehetővé teszi a bejelentkezés letiltását a rendszerbe. Ez a funkció segítséget nyújt az adminisztrátornak, hogy a felhasználókat ideiglenesen "letiltsa" vagy "inaktívvá" tegye a rendszerben anélkül, hogy a felhasználói fiókot ténylegesen törölné. Az adminisztrátornak lehetősége van véglegesen törölni is.

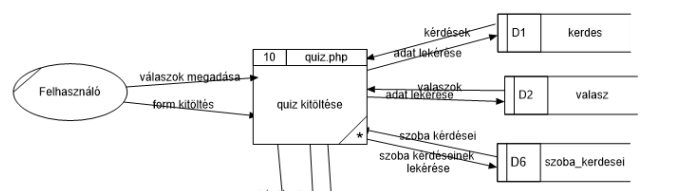
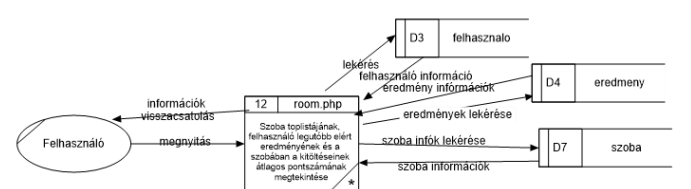
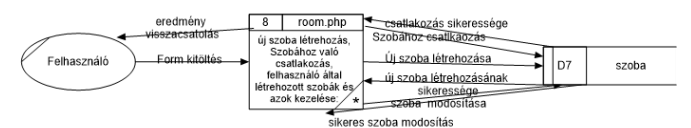
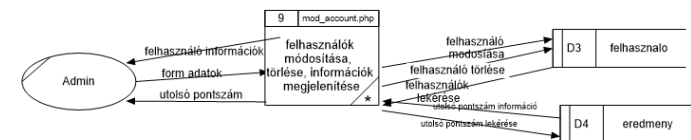
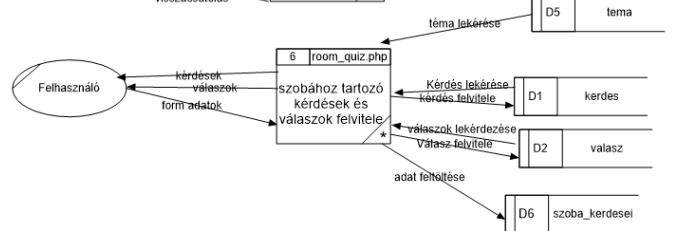
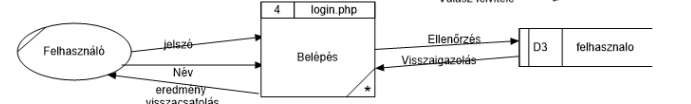
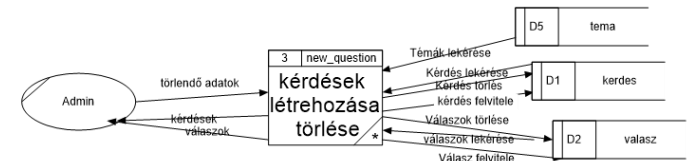
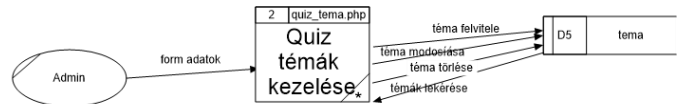
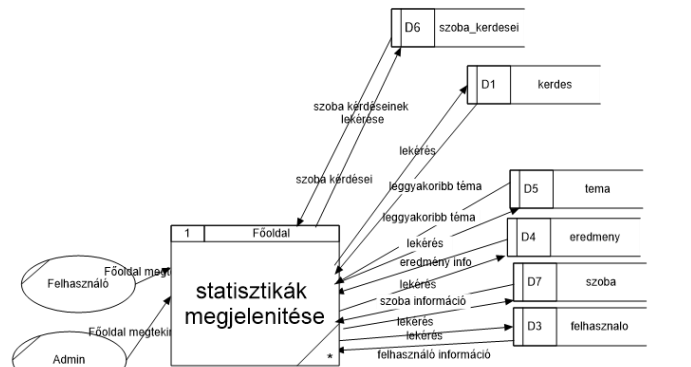
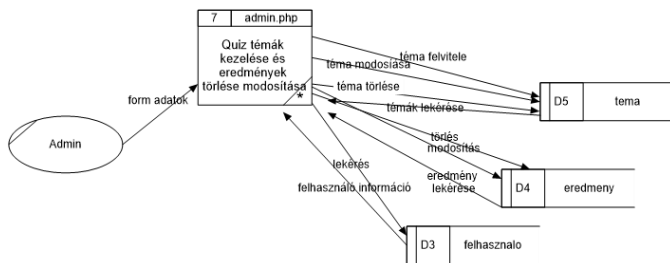
Nem funkcionális követelmények:

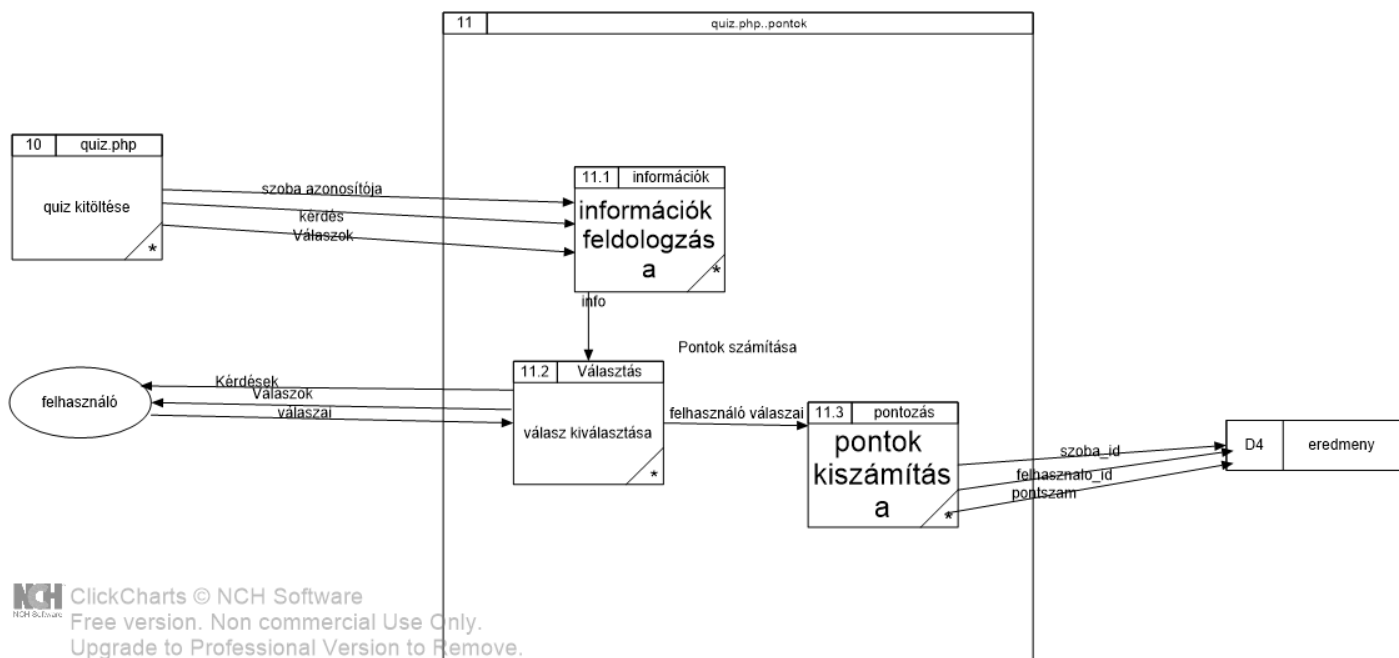
- Megbízhatóság
- Teljesítmény
- Felhasználói felület
- A programnak grafikus felhasználói felületet kell biztosítania a felhasználók számára.
- A felhasználói felületnek ergonomikusnak kell lennie, és az információkat jól átlátható módon kell megjelenítenie.
- A felhasználói felületnek tartalmaznia kell a szükséges gombokat és elemeket, amelyekkel a felhasználók könnyen használhatják a programot.

- Karbantarthatóság
- Biztonság
- A programnak biztosítani kell az adatok biztonságos tárolását és kezelését.
- A programnak lehetővé kell tennie a felhasználók jogosultságainak kezelését, például az adminisztrátori jogosultságokat.
- A programnak megfelelő védelmet kell biztosítani a bizalmas adatokkal szemben, például a jelszavak védelmével.

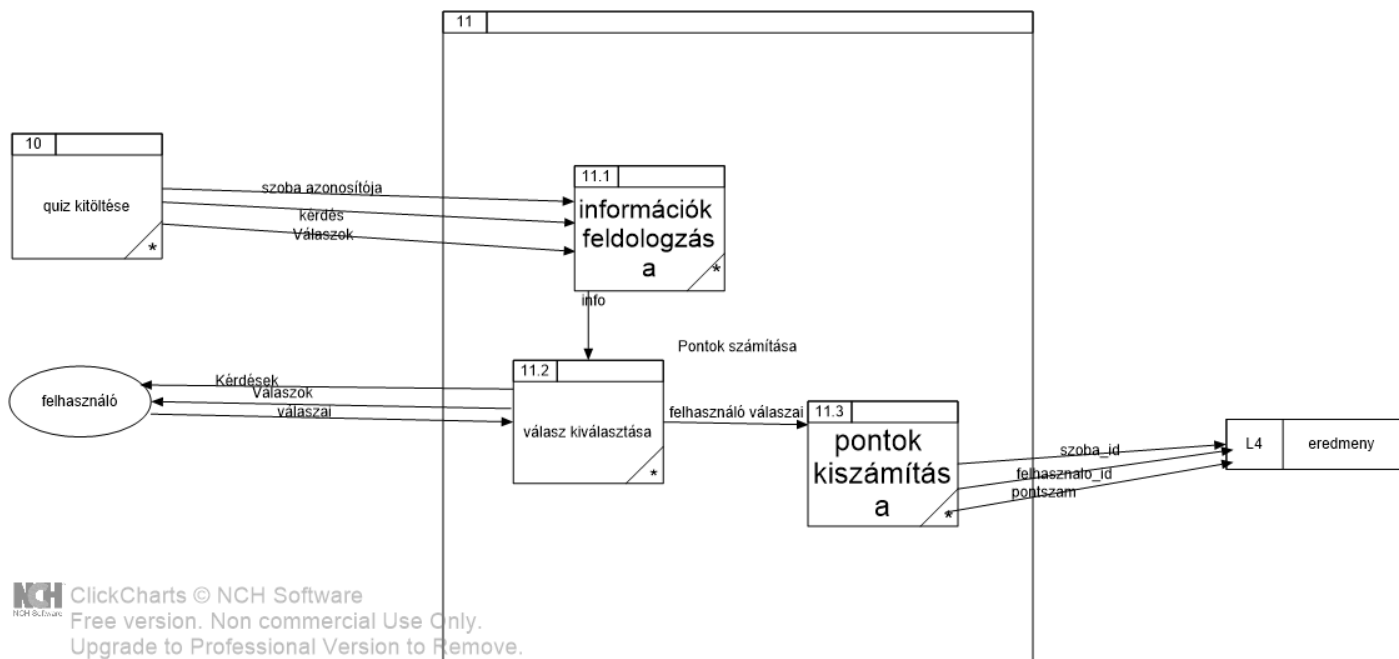
Adatfolyam diagram (DFD):

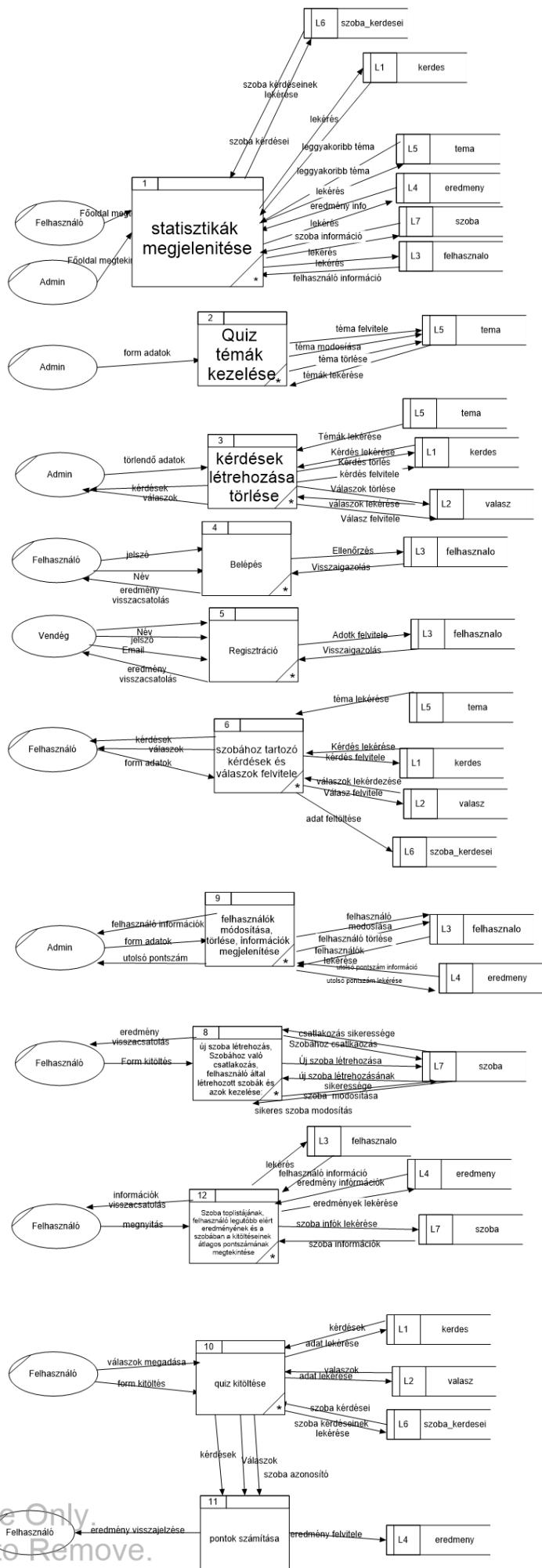
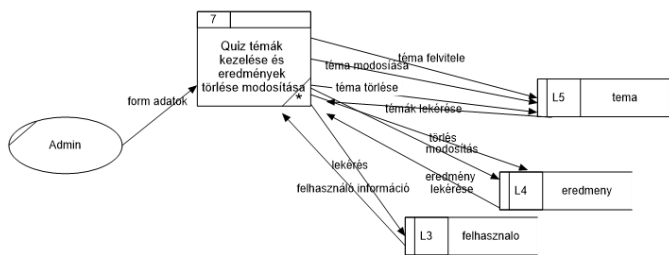
Fizikai



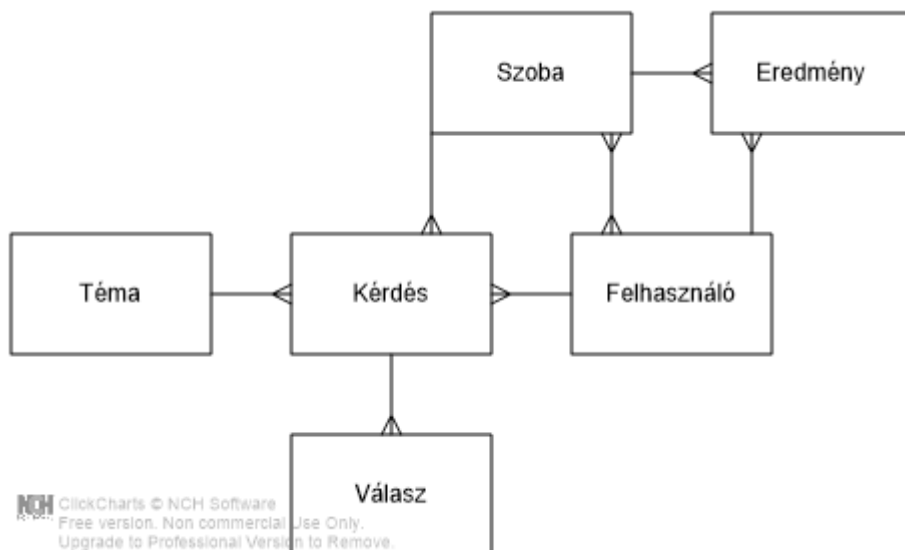


Logikai





Egyedmodell:




```
felhasznalo (id, nev, email, jelszo, admin-e, utolso_aktivitas_datum)
eredmeny (id, szoba_id, felhasznalo_id, pontszam)
kerdes (id, kerdes, tema_id, felhasznalo_id, globalis_kerdes)
tema (id, nev)
valasz (id, kerdes_id, valasz ,helyes_e)
szoba (id, nev, jelszo, felhasznalo_id, utolso_aktivitas_datum)
szoba_kerdesei(id, szoba_id, kerdes_id)
```

eredmenyei(felhasznalo.id , eredmeny.felhasznalo_id)
szobája(felhasznalo.id , szoba.felhasznalo_id)
kerdesei(kerdes.id , szoba_kerdesei.kerdes_id)

Relációs adatelemzés

1. normálforma (1NF):

- felhasználó (id, nev, email, jelszo, admin-e, utolso_aktivitas_datum)
- eredmeny (id, szoba_id, felhasznalo_id, pontszam)
- kerdes (id, kerdes, tema_id, felhasznalo_id, globalis_kerdes)
- tema (id, nev)
- valasz (id, kerdes_id, valasz ,helyes_e)
- szoba (id, nev, jelszo, felhasznalo_id, utolso_aktivitas_datum)
- szoba_kerdesei(id, szoba_id, kerdes_id)

Megjegyzés: Az adatbázis már eleve kiindulópontjában megfelel az 1NF-nek.

2. normálforma (2NF):

- felhasználó (id, nev, email, jelszo, admin-e, utolso_aktivitas_datum)
- eredmeny (id, szoba_id, felhasznalo_id, pontszam)
- kerdes (id, kerdes, tema_id, felhasznalo_id, globalis_kerdes)
- tema (id, nev)
- valasz (id, kerdes_id, valasz ,helyes_e)
- szoba (id, nev, jelszo, felhasznalo_id, utolso_aktivitas_datum)
- szoba_kerdesei(id, szoba_id, kerdes_id)

Megjegyzés: Az adatbázis már eleve kiindulópontjában megfelel az 2NF-nek, hiszen minden táblának van egyetlen elsődleges kulcsa, és nincsenek benne összetett attribútumok.

3. normálforma (3NF):

- felhasználó (id, nev, email, jelszo, admin-e, utolso_aktivitas_datum)
- eredmeny (id, szoba_id, felhasznalo_id, pontszam)
- kerdes (id, kerdes, tema_id, felhasznalo_id, globalis_kerdes)
- tema (id, nev)
- valasz (id, kerdes_id, valasz ,helyes_e)
- szoba (id, nev, jelszo, felhasznalo_id, utolso_aktivitas_datum)
- szoba_kerdesei(id, szoba_id, kerdes_id)

Megjegyzés: Az adatbázis már eleve kiindulópontjában megfelel az 3NF-nek, hiszen az egyes táblákban található attribútumok között nincs tranzitív függőség.

Táblák leírása:

felhasznalo: *felhasznalo* tartalmának leírása.

Név	Típus	Leírás
id	NUMBER	egyedi azonosító
nev	CHAR	felhasználó belépési neve
email	CHAR	Email- cím
jelszo	CHAR	jelszó
admin-e	INTEGER	adminisztrátor-e vagy sem
utolso_aktivitas_dat	TIMESTAMP	Az utolsó lepontozott eredmény idejét menti le

um		vagy a regisztráció dátumát
----	--	-----------------------------

eredmeny: Ez az adattábla tárolja a játékos eredményét egy adott szobában

Név	Típus	Leírás
id	NUMBER	egyedi azonosító
szoba_id	FOREIGN KEY	Szoba azonosítója
felhasznalo_id	FOREIGN KEY	Felhasználó azonosítója
pontszam	INTEGER	pontszámuk

kerdes: Ez az adattábla tárolja az összes kérdést, amelyek az alkalmazásban megjelennek.

Név	Típus	Leírás
id	NUMBER	egyedi azonosító
kerdes	CHAR	kérdés szövege
tema_id	FOREIGN KEY	Téma azonosítója
felhasznalo_id	FOREIGN KEY	Felhasználó azonosítója
globalis_kerdes	NUMBER	egy kérdést bármelyik játékszobában meg lehet jeleníteni

tema: Ez az adattábla tárolja az összes témát, amelyek az alkalmazásban megjelennek.

Név	Típus	Leírás
id	NUMBER	egyedi azonosító
nev	CHAR	Téma megnevezése

valasz : Ez az adattábla tartalmazza az összes választ, amelyeket a felhasználók az egyes kérdésekre adhatnak

Név	Típus	Leírás
id	NUMBER	egyedi azonosító
kerdes_id	FOREIGN KEY	válaszhoz tartozó kérdés azonosítója
valasz	CHAR	válasz szövege
helyes_e	INTEGER	A kérdése vonatkozó helyesség

szoba: Ez az adattábla tartalmazza a szoba információját

Név	Típus	Leírás
id	NUMBER	egyedi azonosító

nev	CHAR	szoba neve
jelszo	CHAR	jelszó
felhasznalo_id	FOREIGN KEY	Azon felhasználó azonosítója, aki a szobát létrehozta
utolso_aktivitas_datum	TIMESTAMP	Az utolsó lementett eredmény idejét menti le vagy a szoba létrehozásának dátumát

szoba_kerdesei : Ez az adattábla tartalmazza az adott szobához tartozó kérdéseket

Név	Típus	Leírás
id	NUMBER	egyedi azonosító
szoba_id	FOREIGN KEY	Szoba azonosítója
kerdes_id	FOREIGN KEY	válaszhoz tartozó kérdés azonosítója

Funkció-szerep mátrix:

Funkciók /Szerepkörök	Felhasználó	Admin	Vendég
Regisztráció			x
Belépés	x	x	
kérdések és válaszok felvitele	x	x	
Globál kérdések és válaszok törlése		x	
új szoba létrehozás	x	x	
Szoba törlése	x		
szobához tartozó quiz készítése	x	x	
quiz kitöltése	x	x	
Felhasználó módosítása/törlése		x	
téma felvitele/törlése/modosítása		x	

Hozzáadott triggererek:

Trigger: update_last_activity_trigger

Leírás:

Ez a trigger minden alkalommal frissíti a felhasznalo tábla utolso_aktivitas_datum mezőjét, amikor egy új rekordot szűrnak be az eredmény táblába, és az új rekordnak van egy felhasznalo_id értéke, amely megegyezik a felhasznalo tábla egyik rekordjának azonosítójával.

Kód:

```
CREATE OR REPLACE TRIGGER update_last_activity_trigger
BEFORE INSERT ON eredmény
```

```

FOR EACH ROW
BEGIN
    UPDATE felhasznalo
    SET utolso_aktivitas_datum = CURRENT_TIMESTAMP
    WHERE id = :new.felhasznalo_id;
END;
/

```

Trigger: update_room_last_activity_trigger

Leírás:

Ez a trigger minden alkalommal frissíti a szoba tábla utolso_aktivitas_datum mezőjét, amikor új rekordot szűrnak be, meglévő rekordot frissítenek vagy rekordot törölnek a szoba_kerdesei táblában, és az adott rekordokhoz tartozó szoba_id érték alapján azonosítja a megfelelő szobát.

Kód:

```

CREATE OR REPLACE TRIGGER update_room_last_activity_trigger
AFTER INSERT OR UPDATE OR DELETE ON szoba_kerdesei
FOR EACH ROW
DECLARE
    v_room_id szoba.id%TYPE;
BEGIN
    -- Szoba azonosítójának lekérdezése az érintett sor
    -- alapján
    IF INSERTING THEN
        v_room_id := :new.szoba_id;
    ELSIF UPDATING THEN
        v_room_id := :new.szoba_id;
    ELSIF DELETING THEN
        v_room_id := :old.szoba_id;
    END IF;

    -- Szoba utolsó aktivitás dátumának frissítése
    UPDATE szoba
    SET utolso_aktivitas_datum = CURRENT_TIMESTAMP
    WHERE id = v_room_id;
END;
/

```

Hozzáadott eljárások:

Eljárás: `get_average_score`

Leírás:

Ez az eljárás egy felhasználó által elért átlagos pontszám lekérdezését végzi a megadott felhasználó azonosítója alapján. Az eredményt az `avg_score` kimeneti paraméterben adja vissza.

Kód:

```
CREATE OR REPLACE PROCEDURE get_average_score(user_id IN
NUMBER, avg_score OUT NUMBER)
IS
BEGIN
    SELECT AVG(pontszam)
    INTO avg_score
    FROM eredmeny
    WHERE felhasznalo_id = user_id;
END;
/
```

Eljárás: `get_all_room_results`

Leírás:

Ez az eljárás az összes szobához tartozó eredményeket kérdezi le.

Kód:

```
CREATE OR REPLACE PROCEDURE get_all_room_results (
    result OUT SYS_REFCURSOR
)
IS
BEGIN
    OPEN result FOR
        SELECT e.szoba_id, e.felhasznalo_id, e.pontszam
        FROM eredmeny e;
END;
/
```

Eljárás: `get_toplist_for_room`

Leírás:

Ez az eljárás az adott szobához tartozó legjobb eredményeket kéri le. A felhasználó nevét és a pontszámot adja vissza egy kimeneti kurzoron keresztül, rendezve a pontszámok csökkenő sorrendjében.

Kód:

```

CREATE OR REPLACE PROCEDURE get_toplist_for_room (
    room_id IN NUMBER,
    result OUT SYS_REFCURSOR
)
IS
BEGIN
    OPEN result FOR
        SELECT f.nev AS felhasznalonev, e.pontszam
        FROM eredmeny e
        JOIN felhasznalo f ON e.felhasznalo_id = f.id
        WHERE e.szoba_id = room_id
        GROUP BY f.nev, e.pontszam
        ORDER BY e.pontszam DESC;
END;
/

```

Összetett lekérdezések:

mod_account.php 61-69

```

SELECT
    felhasznalo.id As felhasznalo_id,
    felhasznalo.email As felhasznalo_email,
    felhasznalo.nev AS felhasznalo_neve,
    TO_CHAR(felhasznalo.utolso_aktivitas_datum,
'YYYY-MM-DD HH24:MI:SS') AS utolso_aktivitas_datum,
    MAX(COALESCE(eredmeny.pontszam, 0)) AS
utolso_pontszam
    FROM felhasznalo
    LEFT JOIN eredmeny ON felhasznalo.id = ered-
meny.felhasznalo_id
    GROUP BY felhasznalo.id, felhasznalo.email, fel-
hasznalo.nev, felhasznalo.utolso_aktivitas_datum

```

A lekérdezés két részből áll:

1. Az első rész egy al-lekérdezés, amely összefűzi a felhasználók adatait és az utolsó pontszámokat.
2. A második részben a fő lekérdezés az al-lekérdezés eredményeit dolgozza fel.

Az `LEFT JOIN` kapcsolatot használva a lekérdezés összekapcsolja a `felhasznalo` és az `eredmeny` táblákat a felhasználók azonosítói alapján. A `GROUP BY` utasítás a felhasználók adatait csoportosítja az azonosítójuk alapján, és kiszámítja az utolsó pontszámot az összes eredmény maximuma alapján.

A `COALESCE` függvényt használva biztosítják, hogy az üres (`NULL`) értékeket 0-ra cseréljék a pontszámokban. A `TO_CHAR` függvény segítségével formázva van az utolsó aktivitás dátuma a megadott formátumra.

home.php 28-33

```
SELECT sk.szoba_id, s.nev AS szoba_neve, COUNT(k.id) AS
kerdesek_szama
      FROM szoba_kerdesei sk
      JOIN kerdes k ON sk.kerdes_id = k.id
      JOIN szoba s ON sk.szoba_id = s.id
      GROUP BY sk.szoba_id, s.nev
      ORDER BY COUNT(k.id) DESC
```

A `szoba_kerdesei` táblát `szoba` és `kerdes` táblákkal összekapcsolva meghatározza az összes szobát, amelyekhez vannak kérdések rendelve. Ezután csoportosítja ezeket a szobákat a szoba azonosítója és neve alapján, és megszámlálja a hozzájuk tartozó kérdések számát. Végül a lekérdezés csökkenő sorrendbe rendezve listázza ezeket a szobákat a kérdések száma szerint.

home.php 73-78

```
SELECT t.nev AS tema_neve, COUNT(*) AS eloformulasok_szama
      FROM tema t
      JOIN kerdes k ON t.id = k.tema_id
      GROUP BY t.nev
      ORDER BY COUNT(*) DESC
      FETCH FIRST 5 ROWS ONLY
```

Ez a lekérdezés az adatbázisból visszanyeri az öt legnépszerűbb témát a kérdések alapján. Az eredmények között szerepel a téma neve és az élőformulák száma, ami a téma alatt lévő aktív kérdések számát jelenti.

home.php 112-116

```
SELECT s.nev AS szoba_neve, COUNT(DISTINCT e.fel-
hasznalo_id) AS aktiv_felhasznalok_szama
      FROM eredmeny e
      JOIN szoba s ON e.szoba_id = s.id
      GROUP BY s.nev
      ORDER BY COUNT(DISTINCT e.felhasznalo_id) DESC
```


Ez a lekérdezés a szobákhoz tartozó aktív felhasználók számát határozza meg. Az eredmények között szerepel a szoba neve és az aktív felhasználók száma, amelyek a szobában kitöltött eredmények alapján kerülnek meghatározásra.

home.php 155-159

```
SELECT s.nev AS szoba_neve, COUNT(e.id) AS kitoltesek_szama
      FROM szoba s
      JOIN eredmeny e ON s.id = e.szoba_id
     GROUP BY s.nev
    ORDER BY COUNT(e.id) DESC
```

Ez a lekérdezés meghatározza a szobákhoz tartozó kitöltések számát az eredmények alapján. Az eredmények között szerepel a szoba neve és a kitöltések száma, ami azt mutatja, hogy egy adott szobában hány eredményt rögzítettek összesen.

home.php 198-204

```
SELECT * FROM (
      SELECT f.nev AS felhasznalo_neve, f.email AS
felhasznalo_email, COALESCE(SUM(e.pontszam), 0) AS ossze-
sitett_pontszam
      FROM felhasznalo f
      LEFT JOIN eredmeny e ON f.id = e.felhasznalo_id
     GROUP BY f.nev, f.email
     ORDER BY COALESCE(SUM(e.pontszam), 0) DESC
   ) WHERE ROWNUM <= 5
```

Ez a lekérdezés meghatározza az öt legaktívabb felhasználót a legtöbb pontszámmal az eredmények alapján. Az eredmények között szerepel a felhasználó neve, e-mail címe és az összesített pontszáma. Csak az első 5 rekordot jeleníti meg.

quiz.php 85.-91

```
SELECT k.tema_id, k.id AS kerdes_id, k.kerdes, v.id AS
valasz_id, v.valasz, v.helyes_e
      FROM szoba_kerdesei sk
      JOIN kerdes k ON sk.kerdes_id = k.id
      JOIN valasz v ON k.id = v.kerdes_id
     WHERE sk.szoba_id = :room_id
     GROUP BY k.tema_id, k.id, k.kerdes, v.id, v.valasz,
v.helyes_e
     ORDER BY k.tema_id
```

Ez a lekérdezés kiválasztja egy adott szoba összes kérdését és az ezekhez tartozó válaszokat. A szoba_kerdesei, kerdes és valasz táblákat összekapcsolva megkapjuk a szükséges adatokat. Az eredményeket csoportosítja tema_id alapján, majd rendezve jeleníti meg a tema_id szerinti sorrendben.

quiz.sql 130-143

```
CREATE OR REPLACE PROCEDURE get_toplist_for_room (  
    room_id IN NUMBER,  
    result OUT SYS_REFCURSOR  
)  
IS  
BEGIN  
    OPEN result FOR  
        SELECT f.nev AS felhasznalonev, e.pontszam  
        FROM eredmény e  
        JOIN felhasznalo f ON e.felhasznalo_id = f.id  
        WHERE e.szoba_id = room_id  
        GROUP BY f.nev, e.pontszam  
        ORDER BY e.pontszam DESC;  
END;  
/
```

Ez a tárolt eljárás visszaadja egy adott szoba legjobbjait a megadott szoba azonosítója alapján. A szoba összes eredményét lekéri, majd ezeket a felhasználó neve és pontszáma szerint csoportosítja. Végül a pontszámok szerint csökkenő sorrendben adja vissza az eredményeket.

new_question.php 77.sor

```
SELECT k.id AS question_id, k.kerdes AS question_text, v.id  
AS answer_id, v.valasz AS answer_text, v.helyes_e AS  
is_correct, k.globalis_kerdes AS is_global FROM kerdes k  
LEFT JOIN valasz v ON k.id = v.kerdes_id
```